FORUM NOKIA

WAP Service Developer's Guide for Nokia Series 30 Phones with WML Browser

Version 1.1; June 09, 2003

NOKIA

Contents

Change History

04 Nov 2002	V1.0	Initial document release.
09 June 2003	V1.1	Template changed, Nokia 3510i and Nokia 8910i information removed.

Disclaimer

The information in this document is provided ”as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

The phone UI images shown in this document are for illustrative purposes and do not represent any real device.

License

WAP Service Developer's Guide for Nokia Series 30 Phones with WML Browser

Version 1.1; June 09, 2003

1 Introduction

The following document provides information and practical examples for developers who want to optimize their WAP services for Nokia Series 30 devices with a WML browser. It is written for service developers as well as anyone involved in creating the wireless information society who needs to know more about service creation on mobile terminals. This guide is not intended to describe WAP in detail, nor is it meant to replace any WAP specification.

1.1 Chapter Topics

Chapter 2 of this document describes several general usability issues concerning WAP service development. Chapter 3 introduces the browser's user interface elements, keys, display, and browsing conventions. Chapter 4 outlines some general WML elements that are mandatory features for WAP 1.2.1, according to the WAP Forum specifications. Chapter 5 introduces additional WAP 1.2.1 features that are supported in Nokia Series 30 phones.

2 Guidelines for Design

The Nokia Series 30 UI is designed to be easy to use and intuitive. The following section offers general guidelines to help developers create browsing applications that will be perceived as usable and easy to understand.

2.1 Applications for Mobile Use

When deciding what information to include in various applications on a mobile terminal, developers should consider the types of situations where the mobile phone will be used. The content of the service should fulfill the needs of the target user group and be optimized for the most common tasks. Because of the mobility of the small display, the user might primarily use the mobile phone when there is no PC access to the Internet, and for quick information. Examples might include speedy access to flight schedules, short news flashes, and access to weather information or checking the temperature at a traveler’s destination. It is less likely that users will use their mobile phones to surf.

2.2 Validating WML

There are several XML validators available that validate documents against WML Document Type Definition. It is recommended that authors validate their WAP pages, because invalid WML is always treated as an error and discarded (that is, it is not shown to the user).

Several XML validators can be found at http://www.wapdevelopers.org/.

2.3 GPRS Usability

GPRS enables efficient use of resources, instant access, fast delivery of information, and innovative charging models.

2.3.1 No changes required by developers

Since the WAP standard was designed to allow content to be delivered over any bearer service, it is not necessary for developers to make changes to their WAP services or applications. This is true even when delivery of the services is enabled over GPRS, 3G, or any other networks.

2.3.2 Cache utilization

Developers should use HTTP proxy to cache WML content whenever the content is accessed through the Internet. The proxy should be placed close to the WAP gateway to minimize the distance for data transfer between the two components. The home page (first page) can alternatively be kept on a local server, close to the WAP gateway.

Most clients have a cache; using it is a good way to improve the user's experience. Cache headers should be sent with documents to ensure that pages are cached on the client if possible. Content developers should not assume that pages are cached by default.

2.3.3 Usability tips

Round trips in the network add a minimum two-second delay to the response time. Therefore the number of round trips should be minimized. The response time of a first page has a strong impact on the user experience, so it is worth optimizing.

2.3.4 Infrastructure optimization

The latency between the WAP gateway and content server should be minimized. Each query traverses this link twice, once as a request and once as a response.

The WAP gateway should be placed close to the GPRS gateway. (The GPRS gateway is technically called a “Gateway GPRS Support Node,” or GGSN.) “Close” in this instance refers to both latency and the probability of packet loss. Lost messages cause additional delays due to HTTP retransmission.

2.4 Site Organization

2.4.1 Doormat pages

It is not recommended to start the site using a doormat page, which serves no other purpose than to perhaps greet the visitor and display a logo. It is better if the user can go to the service directly.

2.4.2 Login

Entering login information can be time consuming with a mobile phone. Login information should be requested only once during the session, not each time after starting navigation from the main page. Case-sensitive checking of user names and passwords should be avoided.

2.4.3 Detecting the browser

When using a modern Web site hosting environment, it is possible to detect the browser as well as the language mode and supply the correct content transparently, without user interaction.

The Nokia Series 30 mobile browser sends a user agent field in the WSP header to the WAP gateway when the content is fetched from the origin server. This user agent field can be used to define the type of browser and then build a logic on the origin server to serve suitable content. From the WAP gateway, it is then delivered inside the HTTP header.

2.4.4 Optimizing for size

Interactive pages should be short; informative pages, long. Having pages that are too short and require many downloads is worse than having close-knit information on one longer page. In particular, interactive pages with forms should not be too long, as users start to lose their feeling of control if the form is more than two screenfuls. Empty rows can be misleading, as users may think there is no more content below.

If, for example, a page contains a story or instructions, the entire content should be on one page. Subtitles that take the user to points within a page help when browsing long, informative pages.

It is difficult to give hard and fast guidelines about how many links should be provided on one link list page. If the links clearly belong together and are easy to browse (one line per link, in alphabetical or other logical order), it is better to provide 30 links on one page than 5 links on six pages.

As for the total downloading time, some studies place an upper limit for acceptable delay on a PC-based browser at 10 to 15 seconds, including all images. It is highly recommended to have a response time (excluding connection time) of less than 10 seconds in a mobile environment.

2.4.5 The first screen

The first (topmost) screenful of any page is the most important. All often-used navigational links, search fields, login screens, and the bulk of the information should reside there. The user can then navigate forward before the rest of the page has been loaded, and will not have to scroll the page.

Wasting the top of the page on banner advertisements or non-informative graphics should be avoided. It is better to place advertisements at the left or right edge.

2.5 Pictures and Tables

Downloading images takes time, and many users may opt to switch off the loading of images for more speed. Authors should try to optimize image size. If there are large pictures on the site, authors should consider using thumbnails for the image index. Color images should be tested for visibility on a monochrome display. Usually, the same image cannot be used for color and monochrome displays.

An alternative text (using the alt attribute of the <img> element) should be provided for images that convey information. A null alternative text (alt="") should be used for images that do not convey information, or that are used for page layout or decorative purposes only.

The table width or height should not exceed the screen size, since scrolling will make the table less readable (e.g., column titles are not visible after scrolling down).

2.6 Efficient Navigation

The service provider must be responsible for navigation as a whole. A consistent navigation method should be provided within the service. Each page should contain links to the main page of the site and to the possible subsection main pages.

The back function can also be provided as a link on the page, but should not be used, for example, on pages that will return the user to a purchasing or confirmation page.

If access keys are used, their definitions should be consistent for the main page links, e.g., "1" always takes the user to the main page of the service.

2.7 Search

A piece of information in the service can be found by moving ahead in a tree hierarchy or by conducting a search. Ideally, both methods should be provided. The tree hierarchy helps the first-time user clarify the service structure, but it may be too slow for an experienced user.

Search results should be optimized to include only a few items; if more are found, the user should have the opportunity to narrow the search criteria.

2.8 Titles and Element Labels

The title describes the contents of the display. Use of titles is recommended in WML. Titles help the user navigate in the application because they remind the user where s/he is within the application. It may be a good idea to start the title with the service's name and keep the total length of the title short. The item previously selected by the user should determine the header text. For instance, the title "Bookmarks" tells the user that the display contains a list of bookmarks in the application and that the options item previously selected was Bookmarks.

Proportional fonts are used in header text, and if the header text is too long, it is cut automatically. Cut titles are usually better than abbreviations, because the user may be confused by unfamiliar abbreviations.

Although short words are recommended for element labels, acronyms that are not well known by the target user group should be avoided. The same label should always be used for the same thing, especially with function labels such as Delete, Remove, Erase, Clear, and Destroy.

2.9 Error Messages and Confirmations

It is especially irritating for users to receive error messages and be asked to correct the input. Services should be very flexible when interpreting input, e.g., allowing several different formats for date and time. If an error message must be shown, it should be helpful and polite, explaining why the action could not be executed, and providing an easy way to correct the situation.

After the user has chosen a command that will carry out an action such as destroying information or moving money, it is essential to provide a confirmation page before executing the action. It should also be easy for the user to check the effects of the action. For example, when deleting an item, it is best to first ask for confirmation and then inform the user that the deletion was executed successfully.

2.10 A Usability Test

It is always good to perform a usability test for new applications. People who have not been involved in the design or development of an application tend to notice potential usability problems that are often not obvious to those who know the design by heart. Usability tests should always be performed as early as possible in the development process. Any necessary changes resulting from the tests can then be implemented within the development timetable. Testers who are representative of the future end users should be used. At the least, tests should be conducted on a small scale if the schedule does not allow for extensive testing.

3 Introduction to the User Interface

The following chapter offers a brief overview of the user interface style in 96 x 65 pixel screen resolution phones. The main focus is on the WAP services element (WAP browser) of the phone. The WAP browser user interface is designed to comply with the user interaction handling known from other applications in the phone.

The user interface style is a combination of specific user interface hardware such as keys and display; user interaction conventions, which include input and output functions and their relationships; and the audio-visual appearance, which is used in the user interaction conventions. This chapter concentrates on the hardware part of the 96 x 65 pixel resolution user interface.

3.1 Nokia Phones with 96 x 65 Pixel Screen Resolution

Nokia Series 30 phones have a WAP browser with enhanced features and increased usability. Predictive text input (T9) can be used in the WAP browser; predictive text input in WAP and a picture viewer are also supported. Nokia Series 30 phones support Push service indication; in addition most of the June 2000 conformance (WAP 1.2.1) phones support WTAI public library and cookies[*]. For more detailed information, see the WAP characteristics table at http://forum.nokia.com.

Nokia 3410	Nokia 3510	Nokia 6310	Nokia 6310i	Nokia 6510
WAP 1.1 + Push	WAP 1.2.1	WAP 1.2.1	WAP 1.2.1	WAP1.2.1
GSM900/1800	GSM900/1800	GSM900/1800	GSM900/1800/1900	GSM900/1800
CSD	CSD, GPRS	CSD, GPRS	CSD, GPRS	CSD, GPRS
Two soft key UI	Two soft key UI	Two soft key UI	Two soft key UI	Two soft key UI

3.2 User Interface Hardware – Keys and Display

3.2.1 Two Soft Key Display

The display is a full dot-matrix display with a display resolution of 96 pixels (horizontal) by 65 pixels (vertical). The display consists of the application area, header area, and an area used for the soft keys.

[*] Excluding Nokia 3410

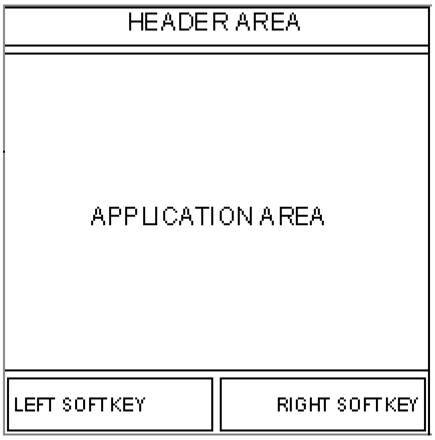

Figure 1: Structure of the browser screen

The application area is 96 pixels (horizontal) by 45 pixels (vertical). It includes four lines of main text and can also be used for displaying graphics, which can be scrolled four ways (picture viewer) until the maximum deck size.

3.2.2 Keys in Two Soft Key User Interface

The **Send/Talk key** works as a selection key.

The **End key** exits a data call/GRPS connection and goes into idle state when pressed twice.

The **two soft keys** are assigned actions that enable the user to manipulate the user interface by making selections and entering, editing, and deleting text. In Nokia 68xx there is also a third soft key, which is used only as profile key.

-The **left soft key** is used as a yes/positive key. It contains options that execute commands and go deeper into the menu structure. Some example functions are Select, OK, and Options.

-The **right soft key** is used as a no/negative key. It contains options that cancel commands, delete text, and go backwards in the menu structure, such as Back, Exit, and Clear.

The **scroll keys (arrow keys)** allow the user to scroll the options or text in the current display.

3.3 Display Fonts

All fonts are proportional fonts. Proportional fonts give a dynamic and minimized width for each character, improve readability of the display texts, and generally allow more characters to be displayed per line. On the other hand, having proportional fonts means that it is difficult to tell how many characters there can be in each line, e.g., compare characters “W” and ”i.”

3.4 Accessing the WAP Browser

The browser is accessed by selecting Services from the menu and entering a home page defined for the currently active settings or a bookmark. For quick access to the browser, the user can make a long key press on the 0 key; the user is then taken to the home page that is defined for the currently active settings set. The access point and home page can be set in the Services -> Settings menu.

3.5 Navigation, Labels, and Titles

The user should be provided with consistent card headers, element titles and do-element labels. The title of a <select> element is used in the header of a selection list or option groups list. The title of an <optgroup> element is used as an option group title and in the header of its selection list. The title of an input element is used in the editor title. The label of a do element is used in the Service options behind the Options soft key.

3.6 Scrolling

The user can move up and down in a card view when it contains elements (hotlinks, input elements, and selection list items) and static text areas between them. This is done by ”jumping” from element to element. Scrolling through long static sections (that is, text lines) is done line by line. The first element of a card is highlighted by default if it is on the first display.

3.7 Picture Viewer

The picture viewer lets users perform horizontal and vertical scrolling of pictures/animations that are too wide/high for the display. The picture viewer can only be activated by selecting the options list item View Picture if the currently active page contains at least one picture/animation. In the picture viewer, pictures/animations are displayed similarly to the way pictures/animations are displayed in the browser display except that only one picture/animation can be viewed at a time.

3.8 Cache

The browser places viewed cards in cache, which allows previously visited cards to be quickly accessed, as there is no need to reload them. The cards stay in cache until it is manually cleared by the user or until the memory is full. See the Nokia WAP Phone Characteristics document for information on cache size in different Nokia WAP phones.

4 WML Elements for the Nokia User Interface

This chapter serves as a guide for using WML in designing services. It provides an overview of graphical user interface elements, that is, the WML elements that are critical from the rendering point of view and their WML capabilities supported by the browser. The document does not include all possible WML elements and attributes. WAP service developers should pay special attention to Sections 4.2: Table and 4.3: Images.

The WML elements are briefly described and demonstrated with screenshots. Code examples include only the essential part related to using the elements, i.e., the document prologue is not shown after Example 1.

4.1 Text Formatting

The contents of a WML card are displayed in the Application area of the display (see Figures 4.1 and 4.2). The card is a container of text, images, tables, and input elements. The order of elements in a card is significant, as they appear on the screen in the same order.

4.1.1 Paragraph and content alignment

The paragraph element, p, enables word wrapping and content alignment. A paragraph always starts on a new line. Normal static text information is always wrapped on a display, unless word wrapping is set off in the browser settings.

Insignificant white space is ignored. If more than one white space separates the constituents of a text, these white spaces are ignored unless the pre element is used. See Section 4.1.4 for more information on the pre element.

The content inside a paragraph can be aligned to the left, center, or right by option; left alignment is displayed by default. Alignment is determined in paragraph attributes (for example, `<p align="right">`). Word wrapping can be set to wrap or nowrap by using the mode attribute in paragraph.

```
<p align="center">
Align &quot;center&quot;
</p>
<p align="right">
Align &quot;right&quot;
</p>
<p align="right">
<img alt="3510i" src="file://C:/USERS/wml/3510i.gif" align="top" hspace="0"
vspace="0"/>
</p>
```

Example 1: Alignment `<align>`

4.1.2 Line break

New lines in text can be defined by a line break. The line break element, `br`, can be used inside other elements as well, as long as it belongs to a paragraph. If a line break is defined after an `a` element, the line break is ignored. If an empty line is needed between two links, it is necessary to add two line breaks between the `a` elements.

4.1.3 Fieldset element

The fieldset element allows the grouping of related fields and text. The basic idea of grouping is to show the grouped items on the same display if possible. If the items cannot be displayed on the same display, the user is not given any notification. Fieldset always starts a new line and ends with a line break.

4.1.4 Pre element

The pre element is supported. When the pre element is used, the content shown is rendered as “preformatted” to the extent possible. This implies that white space is left intact when rendered, that the font in the cards remains the same as for plain text in a card, and that word wrapping is enabled/disabled according to how the browser is set.

4.2 Table

The <table> element is used together with the <tr> and <td> elements to create sets of rows and columns of data, such as text, images, and so on. It is possible to have text, images, and tables on the same card. The cells are shown in bordered rows and columns.

The size of the cells is dynamic, according to the content of the cells. The widest cells affect the width of the other cells in the same column. There is a minimum cell width of 10 pixels in order for the columns to remain discernible. The width of the rows will be adapted so that all can fit. When more cells are defined than can be on the screen, the cells will all be of the defined minimum width. When the content of the cells is too large to fit the cell, its content is truncated and three dots are added.

A maximum of one text, image, or link element can be shown in each cell when no row in the table has been selected. Otherwise text, image, as well as link elements can be shown in a cell. In a text element, if line breaks are defined, several lines can be shown at a time. When no row has been selected, the longest line in a cell defines the width of the cell if the cell can be shown in full width. Otherwise lines of excessive length are truncated and three dots are added. The full content of each cell can be viewed by scrolling.

The title of the table is shown when a row is selected. If the table has no title, the title of the card in which the table is situated is used as a title for the table.

```
<table align="LCC" columns="3">
  <tr>
     <td>Date</td>
    <td>F'cast</td>
    <td>T °C</td>
  </tr>
  <tr>
    <td>
    <anchor title="date">M 6/7
    <go href="#date" method="get" sendreferer="false"/>
    </anchor>
    </td>
    <td>
    <anchor>
    <img alt="rain"src="file://…/rainy.wbmp"
    align="bottom"/>
    <go href="#descr" method="get" sendreferer="false"/>
    </anchor>
    </td>
    <td>25°C</td>
  </tr>
  <tr>
    <td>T 6/8</td>
    <td> <img alt="part cldy" src="file://…/partcldy.wbmp"
align="bottom"/> </td>
    <td>27°C</td>
  </tr>
  <tr>
    <td>W 6/9</td>
    <td> <img alt="cloudy" src="file://…/cloudy.wbmp"
    align="bottom"/> </td>
    <td>24°C</td>
  </tr>
  <tr>
    <td>T 6/10</td>
    <td> <img alt="rainy" src="file://…/rainy.wbmp"
    align="bottom"/> </td>
    <td>28°C</td>
  </tr>
  <tr>
    <td>F 6/11</td>
    <td> <img alt="sunny" src="sunny.wbmp"   align="bottom"/> </td>
    <td>29°C</td>
  </tr>
</table>
```

Example 2: Table element

4.3 Images

The graphics formats Wireless Bitmap (wbmp), static gif (.gif 87a), and animated gif (.gif 89a) are supported. The static graphics area is at maximum 78 pixels wide and 30 pixels high, but vertically and horizontally scrollable until the maximum deck size. However, the animated gif graphics area is at maximum 78 pixels wide and 30 pixels high. Images fitting in the graphics area are centered unless specified otherwise by the service provider. Static images that are too wide are left-aligned and truncated on the right. Images that are too long are top-aligned but scrollable down.

The displaying of images is optional and can be turned off in the browser appearance settings. Images can also be used inside a table and as a link (see Example 4.2 above). An image that functions as a link is represented with a full-line frame around it. Images appear in a separate line than text.

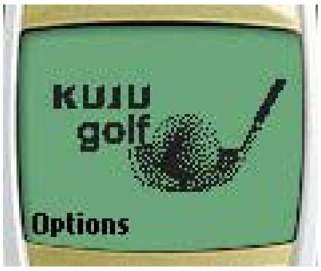

Figure 2: Image in Nokia 6310

Image sizes of a maximum of one line, that is, 10 pixels high, may be shown with two lines of text on the same display above and/or below the image. Similarly, with an image of a maximum of 20 pixels there can be one line of text.. There cannot be text next to an image; the image will always start its own line.

If a requested image does not exist, a small default picture will be shown. Use of the image alt attribute value is recommended.

4.3.1 Animated GIF

Most recent Nokia browsers[1] support image format animated gif (.gif 89a). The maximum number of frames allowed is 13 (note, maximum file size is 2.4 kb).

The animation can be viewed in the browser view. After running the animation in looping mode for 50 loops the animation will stop. The last frame of the animation will be shown as a static gif.

4.4 Links

The anchored link element specifies a hotlink. Links are underlined; in Nokia 3510i links are displayed and underlined in blue. An active link is shown in inverse colors. When a user selects a link, the associated task is performed. When the link is active, the options list of the browser display is extended with the option Select. The user can also select a link by pressing the Send key.

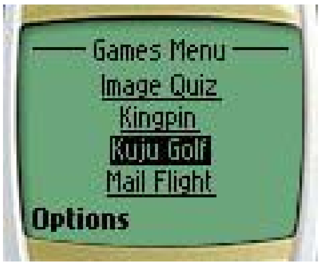

Figure 3: Selected link is highlighted

The a element is a short form of the anchor element, and is bound to a go task without variables. In general it is recommended to use the a element instead of the anchor where possible, to allow more efficient tokenization.

A hotlink is automatically followed by a line break. If a line break is defined after an a element, the line break is ignored. If an empty line is needed between two links, it is necessary to add two line breaks between the a elements. If a hotlink does not have a title or label, the URL address of the link is displayed in its full length, and functions as a normal hotlink.

```
<anchor>text<go href="#card2"/>
</anchor>
```

```
<anchor>
<img alt="golf" src="file://…/golf.wbmp"/>
<go href="#card2"/>
</anchor>
```

Example 3: Image as a link

4.5 Do Element

The do element provides a general mechanism for the user to act upon the current card. The do elements are mapped behind the browser´s options list together with the preloaded options. If a do element contains a prev task, the right soft key is labeled "Back." Use of a prev task is highly recommended in every card because it enables backward navigation. Do elements have a label, i.e., the text of the item in the browser's options list. If the label is available it is used as an option list item. If there is no label, a default text is displayed, according to Table 4.1 below. However, it is highly recommended to use the `label` attribute.

[1] All phones that are introduced in this document support gif image formats in the WAP browser.

Type	Static text
Accept	OK
Prev	Back
Help	Help
Reset	Reset
Options	Options
Delete	Erase
Unknown	Unknown

Table 1: do elements label default text

4.5.1 Prev task

The author of the WAP service is always responsible for the working navigation model inside its own service. If there is a need for back functionality, it should be explicitly defined with the `prev` type do element. The WAP browser does not have any implicit back functionality.

4.5.2 Cardwide and deckwide scope

The do element can be expressed in two ways: in cardwide scope or in deckwide scope in template. Do elements will be stored in the browser's options list together with the preloaded options.

```
<template>
<do type="prev" label="Back">
<prev/>
</do>
</template>
```

Example 4: Deckwide backward navigation, `prev`

```
<do type="prev" label="Back"
<prev/>
</do>
```

Example 5: Cardwide backward navigation

```
            <wml>
<template>
<do name="help-button" type="options" label="Ask Help">
<go href="http://.../help.wml" method="get"/>
</do>
<do type="options" label="Contact">
<go href="http://.../contact.wml" method="get" sendreferer="false"/>
</do>
<do type="prev" optional="false">
<prev/>
</do>
</template>
<card id="card1" title="Homepage" ordered="true">
<p align="left">
Please, register
<br/>
Username:
<input name="username" type="text" title="Username"
emptyok="false"/>
Password:
<input name="password" type="password" title="Password"
emptyok="false"/>
<do type="accept" label="Enter">
<go href="#confirm" method="get" sendreferer="false"/>
</do>
</p>
</card>
<card id="confirm" title="Confirmation" ordered="true"
newcontext="false">
<p align="left">
$(username), thank you for registering.
</p>
</card>
</wml>
```

Example 6: Do element

4.6 WML Input Processing

Input processing lets the user input requested information to the service. There are two kinds of input elements: text fields determined by an input element and selection lists determined by a `select` element. `Option` elements are used to specify a single choice option in a `select` element. `Option` elements can be grouped by using an `optgroup` element (see Example 4.9).

4.6.1 Input elements

The `input` element, which is displayed with brackets, specifies a character entry object. By selecting the input element, the user evokes the editor. The editor is never displayed directly in the card view. The options list of the browser display is then extended with the "Edit" option. The T9 option is enabled in the browser. The user can select the input element by pressing the Send key. The input element consists of a title, brackets, and a value. There can be a default value inside the brackets. The title of the element is only displayed when the editor box is displayed, which is after having selected the input element. The value with the brackets is always wrapped to the next line. If the value cannot fit into one line, the end of the content is truncated and three dots are added.

4.6.2 Direct input access

The user can also select the input element by pressing any numeric key when the edit box appears and the pressed key is already placed in the edit box, i.e., the user can just start inputting characters and the edit box appears automatically.

When T9 (predictive text input) is activated, the soft keys are "Options" and "Back" when the editor is empty. Otherwise the soft keys are "Options" and "Clear." This also applies if T9 is deactivated. Input fields can be specified by the input element's type and format attributes.

```
<template>
<do type="previous" label="Back" optional="false">
<prev/>
</do>
</template>
<card id="card1" title="Homepage">
<p align="left">
Please, register
<br/>
Username:
<input name="username" type="text" title="Username"/>
Password:
<input name="password" type="password" title="Password"/>
<do type="accept" label="Enter" optional="false">
<go href="#confirm" method="get" sendreferer="false"/>
</do>
</p>
</card>
<card id="confirm" title="Confirmation" ordered="true">
<p align="left">
$(username), thank you for registering.
</p>
</card>
</wml>
```

Example 7: Text input `<input>`

4.6.3 Select element

Selection lists are elements that specify a list of options for the user to choose from. There are two kinds of selection lists supported: single-choice and multiple-choice lists. The user can select multiple choice, if the `multiple` attribute is set as `true` (selected items are marked with a selection symbol). The user can highlight and select the selection list item on a card. The options list of the browser display is extended with "Select." The user can also select the selection list by pressing the Send key. That same item will be highlighted and displayed with a selected icon in front of it once the user re-enters the list during the same browsing session.

A selection list item on a card looks similar to an `input` element item; it consists of a title, brackets, and a value. There is always a default value inside the brackets in a single selection list (in a multi-selection list there are by default three dots between the brackets unless specified otherwise by the service provider). If there are no options in the selection list, three dots are displayed between the brackets. The title is not displayed in the card itself. The value with the brackets is always wrapped to the next line, which means that there will never be text and a value with brackets on the same line. If the value cannot fit onto one line, the remaining part of the content is truncated and replaced with three dots.

The title of the selection list is the first item of the card containing the selection list. If the value is a value of a multi-selection, the selections are separated with commas. If the value cannot fit into one line, the end of the content is truncated and replaced with three dots. The user can view the value between brackets in full by pressing the # key. Until the # key is released, the content that can fit on the screen is viewed. The soft keys are hidden while the content is viewed in full. The layout in full view is the same as the one selected by the user for browsing.

The `option` element specifies a single choice option in a select element. Options can be grouped with an `optgroup` element, which specifies a group of choice items in a select element. An `optgroup` element is displayed with an icon in front of the title of the `optgroup` element. An `optgroup` element can be used in single selection lists as well as multiple selection lists. Only in multiple selection lists it is possible to have option groups that allow multiple selections.

```
<wml>
<card id="phones" title="Information Request">
<p align="left">
Send me more information about:
<select name="models" title="Products" multiple="true">
  <optgroup title="Phones">
  <option value="Nokia 7650">
    7650
  </option>
  <option value="Nokia 9210">
    9210
  </option>
  <option value="Nokia 6310">
    6310
  </option>
  </optgroup>
  <optgroup title="Accessories">
  <option value="Battery">
    Battery
  </option>
  <option value="Desktop stand">
    Desktop stand
  </option>
  <option value="Charger">
    Charger
  </option>
  <option value="Headset">
    Headset
  </option>
  </optgroup>
</select>
Please, provide your
<br/>
Name:
<input name="name" type="text" title="Name" emptyok="false"/>
</p>
<do type="accept" label="Confirm" optional="false">
<go href="#confirm" method="get" sendreferer="false"/>
</do>
<do type="prev" optional="false">
<prev/>
</do>
</card>
</wml>
```

Example 8: Select and optgroup

5 Additional WAP 1.2.1 features

5.1 Push

Nokia Series 30 phone is able to receive Push Service Indication (SI) type service messages. Push service messages are an ideal way for service providers to activate users, since they can be used to notify users of updated content or to alert them about information in their areas of interest (e.g., new e-mails, changes in stock prices, flight schedule changes or news headlines). Service messages indicate downloadable content. The user can view indicated content by using the Service Inbox and the WAP browser.

5.1.1 Service inbox

The service provider can push a Service Indication message to the phone, which is then saved to the Service Inbox in the Services menu. The user can view indicated content afterwards by using the Service Inbox. In the Service Inbox, the received service messages can be viewed, deleted, downloaded, or moved to another folder. Users can define in the Service message settings whether to receive service messages or not.

Service messages are sorted in the Service Inbox into the following order:

1. All unread Service Indications appear highest on the list and are grouped according to the applied priority level, starting with “signal-high,” then “signal-medium,” and finally “signal-low” (see Section 5.1.2 for more information about the priority levels). The messages in each group are shown in chronological order according to the receiving time.

2. All read service messages appear below unread service messages and are shown in chronological order according to when the Service Indications were created.

5.1.2 Service Indication

At the least, a Service Indication should contain text that is displayed to the user and a URL address of content that may be dynamically generated for each push message in the application server. The Service Indication can contain the following information:

- Text
- URL address
- Creation date
- Creation time
- Expiration date
- Expiration time

Each Service Indication has a Service Indication ID, which is used to uniquely identify different kinds of Service Indications. Service providers set the Service Indication IDs for messages. The ID allows reception of multiple Service Indications with the same URL address and priority level. If the received Service Indication has the same ID as the one stored in memory, the received Service Indication will replace the stored one only if it is newer.

Service providers can determine the priority level of the message. Priority levels are signal-high, signal-medium, signal-low, signal-none, and delete. Priority level defines how the received messages are ordered in the Service Inbox and also whether the user is notified about received Service Indication. The defined priority level has the following impact:

- signal-high: Notification with an SMS alert tone is presented.
- signal-medium: Notification with an SMS alert tone is presented.
- signal-low: No indication of received Service Indication is displayed to the user. The Service Indication is silently stored in the Service Inbox.
- signal-none: Service Indication is automatically discarded. The user is not notified.
- delete: If a Service Indication already exists, the Service Indication with the same ID in the Service Inbox is deleted. If there is no Service Indication with the same ID in the Service Inbox, the received Service Indication is discarded. The user is not notified.
- If no attribute has been specified, Service Indications will be received as signal-medium.

5.2 Wireless Telephony Applications Interface (WTAI) Public Library

The WAP browser supports three Wireless Telephony Applications Interface (WTAI) public library functions: making a call while browsing, sending DTMF tones, and saving numbers and names to the phone book. These functions, MakeCall, SendDTMF, and AddPBEntry, can be used anywhere in the WML code in the same way as when URI or script references are used for normal non-WTAI references.

5.2.1 Making a phone call

The MakeCall function is used for making a phone call directly from the WML browser application. A typical example of the usage of MakeCall is when the user selects an anchored link, which initiates a phone call. When a MakeCall function is executed, the user is asked to confirm that the dial-up is to be made. If the phone number that is about to be called does not exist in the phone book, a confirmation query with the phone number is displayed. If the number to be called exists in the phone book, a confirmation query is displayed with the name fetched from the phone book. If the user accepts making the phone call, another confirmation query is displayed with text "Quit browsing?"

If the user accepts both queries, the browser session will be terminated and the browser application will close down. The call will then be initiated. When the call is terminated, the phone is in idle state. If the user rejects the latter confirmation query, the call will be initiated from within the browser display and the user can browse cached content during the call. No in-call options such as Mute, Hold, etc., are accessible from within the browser application. The options list belonging to the browser display remains unaffected by the fact that a call is active.

5.2.2 Incoming call while browsing

Normal call handling applies when a call is initiated from the browser part (e.g., the called party is busy, call barring is enabled, etc.).

5.2.3 Sending DTMF tone

The user can send DTMF tones through the network by executing a WTAI function called SendDTMF. Just like MakeCall, a typical way of using SendDTMF is to assign the SendDTMF function to an anchored link. The user must have a voice call active. When the DTMF tone has been sent, the user returns to the card/deck where s/he initiated the sending. The voice call remains active.

5.2.4 Adding a phone book entry

The AddPBEntry function offers an easy way to store a phone number and a corresponding name onto the phone book application from a WML card. The service provider provides the name and number. When the user activates an AddPBEntry function and the service provider has declared a name and a number, the phone displays a confirmation query with the text "Save name." When the user accepts the confirmation query, another confirmation query with the text "With number +44 123 123 4567" is displayed and the number is stored in the phone book application. When saving is complete, the currently active card is displayed again. Note that the user does not have to be online or have any voice calls active in order to use the AddPBEntry function.

5.3 Cookies

Cookies enable the storage of data such as user information to ease browsing by reducing the amount of information the user must enter. For example, login can be configured dynamically so that it is displayed only if the application cannot identify the user.

Cookies in cached files are ignored.

5.4 Accesskey

Service providers have the possibility of using the accesskey attribute of the a element and the anchor element. This allows users to open links with a long press of the associated access keys. Keys allowed for access key use are the 0-9 number keys. The service provider should indicate to the user which keys to press in order to open the desired links; this can be accomplished by adding the label of the key to the link concerned, e.g., if the service provider associates the “2” key to a link, the service provider should add the number “2” to the label of the link.

```
<a accesskey="2" href="phone.wml">(2)a </a>
```

Example 9: Using accesskey "2"

Only one link can be associated with each access key. A key will also open its corresponding link if the service provider has assigned it as an access key but does not indicate this to the user. If the user tries to open a link by using a key that the service provider has not associated with any link, the key will not react to the action of the user. An access key has an effect on a link even if the link is not highlighted. The link chosen by the user is highlighted after the corresponding access key has been given a long press. This is visible while the terminal is fetching the content of the link. A long press of an access key also has an effect when a card contains both access keys and a highlighted input element. An input element can be activated by an associated access key through a long press in the same way as anchored links (described above).

5.5 Content Download over WAP

Content download is a technology that is used to deliver digital content such as entertainment and business applications to mobile terminals. Another important application is for personalizing terminals according to a user’s preferences and lifestyle. Content downloading can be initiated by an end user or by a network application.

The content download mechanism between the mobile terminal and content source is based on open technologies such as HTTP (and WSP) and Java Community Process technology. This technology allows the creation of an open, multivendor market for terminal vendors as well as for network server and broker vendors.

For more information about content download, please go to http://www.forum.nokia.com/.

5.6 Segmentation and Reassembly (SAR)

Some of the Series 30 phones support Segmentation and Reassembly (SAR) functionality, which is an optional feature in the Wireless Transport Protocol (WTP) layer defined by the WAP Forum. SAR is used to split large WTP messages into a number of smaller datagrams and selectively retransmit failed datagrams. The use of SAR improves over-the-air efficiency for downloading large messages to the phone and makes content download (e.g., Java MIDlet download) and MMS transfer faster and more reliable.

The use of SAR enables the transmission of large data files through WAP to the mobile phone. To use SAR, it must be supported by both of the communicating parties, typically the WAP gateway and the WAP browser. WAP service developers do not need to take any actions to enable SAR in their service development. Even though phones supporting SAR allows downloading of large files, it should be noted that other elements in the network (WAP gateways) might restrict the downloadable file sizes. Nokia WAP gateways let through files up to 100 KB but the limit might be smaller than that in some gateways. Another factor to keep in mind is that SAR can only be used in connection-oriented (permanent) mode.

Further information about SAR functionality can be found in the "Segmentation and Reassembly in the Wireless Transaction Protocol" document, which is available at the Forum Nokia Web site, http://forum.nokia.com.

6 Terms and Abbreviations

Term or Abbreviation	Description
Card	A single WML unit of navigation and user interface.
Deck	A collection of WML cards that is the smallest downloadable unit. You cannot download a single card in a deck; you must download the entire deck. Service developers should be aware of this, and not create large applications in a single deck.
Do element	A way of defining the binding between events (such as a user selection) and a task (for example, to go somewhere).
DTMF tone	Standard tone used to transmit numeric data via telephone. Generated when numeric keys are pressed.
GPRS	General Packet Radio Service. Mobile service that gives a packet switched access over GSM to external data networks with high peak transfer capacity.
GUI	Graphical User Interface.
HTML	Hypertext Markup Language. The markup language used for publishing hypertext on the World Wide Web.
HTTP	Hypertext Transfer Protocol. An application-level protocol for transferring HTML documents in the Internet.
Hyperlink	A link within a document or card that allows quick navigation to another document (card).
Input element	A markup element that allows for interaction with the user, that is, letting the user input textual values. These can be used for inputs to locally stored scripts, or for parsing to origin servers.
Multiselection list	A selection list, which allows the user to scroll and select several (or none) options to be active.
Navigation	The concept of moving between menu items, decks, and cards.
Proportional font	A font in which different letter widths do not necessarily have the same number of pixels. "W" and "I" are good examples.
SAR	Segmentation and Re-assembly. A functionality used to split large WTP messages into a number of smaller datagrams and selectively retransmit failed datagrams.
Selection list	List of user-selectable options.
Soft key	The phone has one or more function keys whose function depends on the state of the phone. These keys are called soft keys because there is an associated text on the display.
UI	User Interface.
URL	Uniform Resource Locator. String that identifies resources in the Web.
W3C	World Wide Web Consortium. An organization that develops interoperable technologies for Web.
Word wrapping	Formatting rule for text appearance.
WML	Wireless Markup Language. Markup language based on Extensible Markup Language (XML), which is intended for

	use in specifying content and user interface for narrowband devices, including cellular phones and pagers.
WSP	Wireless Session Protocol. A layer of WAP stack, which is optimized for low bandwidth bearer networks with long latency.
WTAI	Wireless Telephony Applications Interface. Feature that provides the means to create telephony applications by using a WAE user-agent with the appropriate WTAI function libraries.

7 References

GPRS-related documentation	http://www.mobilegprs.com
Nokia Mobile Internet Toolkit 4.0	http://forum.nokia.com
Nokia Phone User's Guides	http://www.nokia.com
Segmentation and Re-assembly-related documentation	http://forum.nokia.com
WAP Push-related documentation	http://forum.nokia.com
Wireless application specifications	http://www.openmobilealliance.org

Appendix A. Wireless Application Protocol (WAP)

Introduction

The Wireless Application Protocol (WAP) is a set of protocols that allows the development of applications and services for use with mobile phones and other mobile devices. The WAP Forum maintains these protocols and their related standards and specifications.

The WAP Forum consists of a number of high-tech companies from the information technology, software, and telecommunication industries. The objectives of the WAP Forum are to:

- Bring Internet content and advanced data applications to digital mobile phones
- Create a global wireless protocol specification that works across different wireless network technologies
- Enable the creation of content and applications that scale across a wide range of bearer networks and device types
- Embrace existing standards and technology wherever possible

WAP Architecture

WAP and Internet architecture

In order to apply as much leverage to the existing Internet standard as possible, the WAP stack closely follows the Internet model. This is illustrated in Figure 1A.

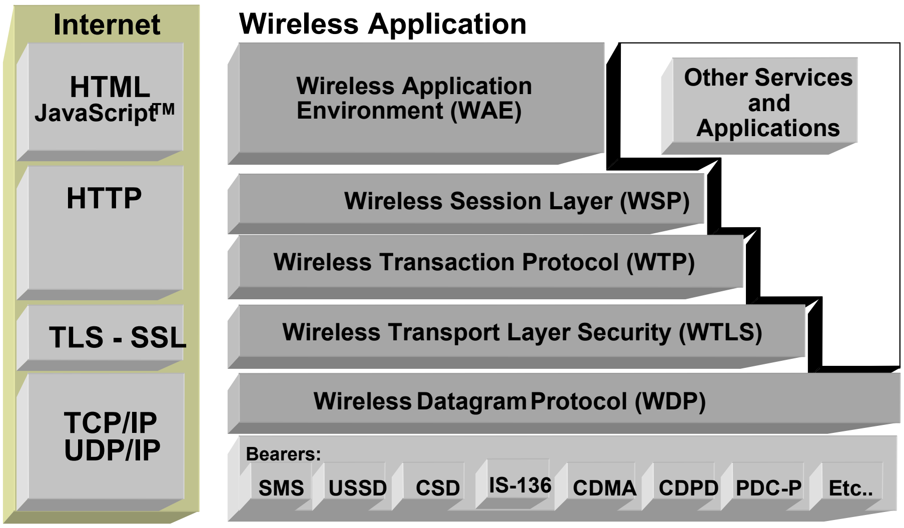
Internet
HTML
JavaScript™
HTTP
TLS - SSL
TCP/IP
UDP/IP
Wireless Application
Wireless Application Environment (WAE)
Other Services and Applications
Wireless Session Layer (WSP)
Wireless Transaction Protocol (WTP)
Wireless Transport Layer Security (WTLS)
Wireless Datagram Protocol (WDP)
Bearers:
SMS
USSD
CSD
IS-136
CDMA
CDPD
PDC-P
Etc..

Figure 1A: Comparison of Internet and WAP architecture

This layered architecture allows applications to use the features of the WAP stack through well-defined interfaces. This close link to the Internet architecture allows developers to use their existing knowledge and expertise when developing applications for mobile terminals.

Wireless Application Environment (WAE)

The Wireless Application Environment (WAE) is a general-purpose application environment that uses a combination of Internet and mobile terminal technology. It provides a framework for the development of applications on a mobile terminal. WAE contains support for the following functionality:

- Wireless Markup Language (WML) - a lightweight presentation language, similar to Hypertext Markup Language (HTML) but optimized for use with mobile terminals.
- Wireless Markup Language Script (WMLS) - a lightweight script language, similar to JavaScript™.
- Wireless Telephony Application / Interface (WTA / WTAI) - telephony services and programming interfaces.
- Content formats - defined data formats, such as vCard and vCalendar.

Wireless Session Protocol (WSP)

The Wireless Session Protocol (WSP) provides the application layer of WAP with an interface for two session services. The first is a connection-oriented service that operates above the transaction layer protocol. The second is a connectionless service that operates above a secure or non-secure datagram service. The WSP is optimized for low bandwidth bearer networks with long latency.

Wireless Transport Protocol (WTP)

The Wireless Transport Protocol (WTP) runs on top of the datagram service and provides a lightweight transaction-oriented protocol, suitable for use in mobile terminals. WTP operates over secure or non-secure wireless datagram networks.

Wireless Transport Layer Security (WTLS)

Wireless Transport Layer Security (WTLS) is based on the industry standard Transport Layer Security (TLS) and is optimized for use over narrowband communication channels. WTLS may be used for secure communication between terminals, and applications can selectively enable WTLS features.

Developing Applications for Mobile Browsing

Application developers can use the principles of WAP to develop new services or adapt existing Internet applications for use with mobile terminals. Applications are written in Wireless Markup Language (WML) and WMLScript, and stored on either a normal Web server (origin server) or directly on the WAP gateway. The content stored on the Web server is accessible from mobile terminals via the cellular network and a WAP gateway or proxy.

The proxy server acts as a gateway between the cellular network and the Internet or Intranet. The data sent between the origin server and the handset is binary encoded to optimize transmission over the narrow bandwidth of the cellular network. Note that the content stored on the Web server might be in either textual or binary format. When the WAP gateway fetches textual content, it automatically compiles this to the encoded format to minimize network load.

Figure 2A shows the network-related elements required for developing and offering services to mobile users.

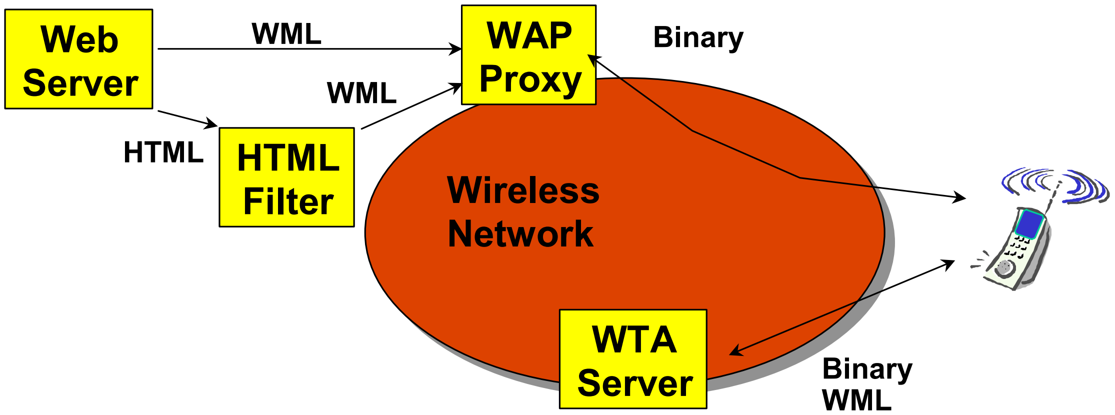

Figure 2A: Elements related to mobile browsing

Web server

The Web server stores the applications written in WML. Alternatively, existing HTML applications can be used.

HTML filter

Any applications written in HTML will have to be converted to WML before they are sent to the mobile terminal. This HTML filter may form part of the Web server or the WAP proxy.

WAP proxy

The WAP proxy acts as the gateway between the cellular network and the Internet or Intranet. It encodes the information, which is to be sent to the mobile terminal in binary format and decodes information sent from the mobile terminal.

WTA server

The WTA server handles network-specific applications. These applications are not discussed in this document.

Additional Information

Further information on WAP is available at http://www.openmobilealliance.org/.

Build Test Sell

Developing and marketing mobile applications with Nokia

1

Go to Forum.Nokia.com

Forum.Nokia.com provides the tools and resources you need for content and application development as well as the channels for sales to operators, enterprises, and consumers.

Forum.Nokia.com

2

Download tools and emulators

Forum.Nokia.com/tools has links to tools from Nokia and other industry leaders including Borland, Adobe, AppForge, Macromedia, Metrowerks, and Sun.

Forum.Nokia.com/tools

3

Get documents and specifications

The documents area contains useful white papers, FAQs, tutorials, and APIs for Symbian OS and Series 60 Platform, J2ME, messaging (including MMS), and other technologies. Forum.Nokia.com/devices lists detailed technical specifications for Nokia devices.

Forum.Nokia.com/documents
Forum.Nokia.com/devices

4

Test your application and get support

Forum Nokia offers free and fee-based support that provides you with direct access to Nokia engineers and equipment and connects you with other developers around the world. The Nokia OK testing program enables your application to enjoy premium placement in Nokia's sales channels.

Forum.Nokia.com/support
Forum.Nokia.com/ok

5

Market through Nokia channels

Go to Forum.Nokia.com/business to learn about all of the marketing channels open to you, including Nokia Tradepoint, an online B2B marketplace.

Forum.Nokia.com/business

6

Reach buyers around the globe

Place your applications in Nokia Tradepoint and they're available to dozens of buying organizations around the world, ranging from leading global operators and enterprises to regional operators and XSPs. Your company and applications will also be considered for the regional Nokia Software Markets as well as other global and regional opportunities, including personal introductions to operators, on-device and in-box placement, and participation in invitation-only events around the world.

Forum.Nokia.com/business